

3 a.**3.a.i.**

The purpose of my program is to provide the user with information about a U.S. state. This app can help with memorization or can just be used as a way to learn something new.

3.a.ii.

The video shows the user selecting the state they want to learn about and then they are directed to a screen showing general information about that state. The user also selects from a dropdown of more information and they click to view the state flag.

3.a.iii.

The input is what state the user selects and what information the user wants from the more information drop down on the general information screen. Both if these items are displayed on the general information screen.

3 b.**3.b.i.**

```
setText("stateOutput", stateName [index]);
setText("statesOutput", stateName [index]);
setText("populationOutput", population [index]);
setText("nicknameOutput", nickname [index]);
setText("capitalOutput", capital [index]);
setText("areaOutput", area [index]);
//setProperty("mapOutput", "image", map [index]);
setProperty("flagOutput", "image", flag [index]);
```

3.b.ii.

```
function filter() {
  for (var i = 0; i < stateList.length; i++) {
    if (stateNumList[i] == 1 || 2 || 3 || 4) {
      appendItem(stateName, stateList[i]);
      appendItem(nickname, nicknameList[i]);
      appendItem(population, populationList[i]);
      appendItem(website, websiteList[i]);
      appendItem(map, mapList[i]);
      appendItem(capital, capitalList[i]);
      appendItem(area, areaList[i]);
      appendItem(flag, flagOutput[i]);
      appendItem(poverty, povertyOutput[i]);
      appendItem(house, houseOutput[i]);
      appendItem(admission, admissionOutput[i]);
    }
  }
}
```

3.b.iii.

The name of the list above is "stateList"

3.b.iv.

The data in the list contains all the information the user will see on the general information screen like State name, Flag, Population, Capital, and Population. This list also contains the information displayed in the “More information” dropdown. Along with all the other information this list is responsible for storing the state flags.

3.b.v.

Without the use of a list the program code would be extraordinarily complex, all the data would have to be coded independently, taking a very long time. The code would be very inefficient without the use of a list therefore it would take much longer to run. Troubleshooting would also be much harder without a list because there is much more to go wrong, and it would be very difficult to find the issue.

3 c.

3.c.i.

```

49
50 //Search Engine
51 function updateScreen() {
52     var index = "";
53     var state = getText("stateInput");
54     if (state.includes("Alabama")) {
55         index = 0;
56     } else if (state.includes("Alaska")) {
57         index = 1;
58     } else if (state.includes("Arizona")) {
59         index = 2;
60     } else if (state.includes("Arkansas")) {
61         index = 3;
62     } else if (state.includes("California")) {
63         index = 4;
64     } else if (state.includes("Colorado")) {
65         index = 5;
66     } else if (state.includes("Connecticut")) {
67         index = 6;
68     } else if (state.includes("Delaware")) {
69         index = 7;
70     } else if (state.includes("Florida")) {
71         index = 8;
72     } else if (state.includes("Georgia")) {
73         index = 9;
74     } else if (state.includes("Hawaii")) {
75         index = 10;
76     } else if (state.includes("Idaho")) {
77         index = 11;
78     } else if (state.includes("Illinois")) {
79         index = 12;
80     } else if (state.includes("Indiana")) {
81         index = 13;
82     } else if (state.includes("Iowa")) {
83         index = 14;
84     } else if (state.includes("Kansas")) {
85         index = 15;
86     } else if (state.includes("Kentucky")) {
87         index = 16;
88     } else if (state.includes("Louisiana")) {
89         index = 17;
90     } else if (state.includes("Maine")) {
91         index = 18;
92     } else if (state.includes("Maryland")) {
93         index = 19;
94     } else if (state.includes("Massachusetts")) {
95         index = 20;
96     } else if (state.includes("Michigan")) {
97         index = 21;
98     } else if (state.includes("Minnesota")) {
99         index = 22;
100     } else if (state.includes("Mississippi")) {
101         index = 23;
102     } else if (state.includes("Missouri")) {
103         index = 24;
104     } else if (state.includes("Montana")) {
105         index = 25;
106     } else if (state.includes("Nebraska")) {
107         index = 26;
108     } else if (state.includes("Nevada")) {
109         index = 27;
110     } else if (state.includes("New Hampshire")) {
111         index = 28;
112     } else if (state.includes("New Jersey")) {
113         index = 29;
114     } else if (state.includes("New Mexico")) {
115         index = 30;
116     } else if (state.includes("New York")) {
117         index = 31;
118     } else if (state.includes("North Carolina")) {
119         index = 32;
120     } else if (state.includes("North Dakota")) {
121         index = 33;
122     } else if (state.includes("Ohio")) {
123         index = 34;
124     } else if (state.includes("Oklahoma")) {
125         index = 35;
126     } else if (state.includes("Oregon")) {
127         index = 36;
128     } else if (state.includes("Pennsylvania")) {
129         index = 37;
130     } else if (state.includes("Rhode Island")) {
131         index = 38;
132     } else if (state.includes("South Carolina")) {
133         index = 39;
134     } else if (state.includes("South Dakota")) {
135         index = 40;
136     } else if (state.includes("Tennessee")) {
137         index = 41;
138     } else if (state.includes("Texas")) {
139         index = 42;
140     } else if (state.includes("Utah")) {
141         index = 43;
142     } else if (state.includes("Vermont")) {
143         index = 44;
144     } else if (state.includes("Virginia")) {
145         index = 45;
146     } else if (state.includes("Washington")) {
147         index = 46;
148     } else if (state.includes("West Virginia")) {
149         index = 47;
150     } else if (state.includes("Wisconsin")) {
151         index = 48;
152     } else if (state.includes("Wyoming")) {
153         index = 49;
154     } else {
155         setProperty("errorMessage", "x", 60);
156         index = "";
157     }
158

```

```

.67
.68 //Extra Info
.69 onEvent("lookupOutput", "click", function () {
.70     var moreInfo = getText("More_Info");
.71
.72
.73     if (moreInfo.includes("Website")){
.74         setProperty("infoOutput", "text", website [index]);
.75     } else if (moreInfo.includes("Percent in Poverty")){
.76         setProperty("infoOutput", "text", poverty [index]);
.77     } else if (moreInfo.includes("Median Household Income")){
.78         setProperty("infoOutput", "text", house [index]);
.79     } else if (moreInfo.includes("Admission Year")){
.80         setProperty("infoOutput", "text", admission [index]);
.81     }
.82
.83
.84
.85
.86 }
.87 );
.88 }
.89

```

3.c.iii.

The above procedure shows the program receiving the state input from the user and updating the state screen with the correct information for the users desired state.

3.c.iv.

Our program uses else if statements to narrows down what state the user wants to see information about. It is definitely not the most efficient way of solving this problem, but for our program it works fine. The user selects the state they want to learn about then the user presses go. The algorithm then works through a series of else if statements checking to see if the state matches with the user inputs and moving on to the next it it's not a match. This continues until it reaches the state that corresponds with the user's input.

3 d.**3.d.i.**

First call:

The first call from the stateInput function is the call to update the screen, set off by pressing the go button.

Second call:

The second call from the stateInput function is the call for more information, this is set off by selecting an option from the drop down on the general information screen then clicking the Go! button.

3 d.ii.

Condition(s) tested by first call:

The user selects their desired state from the drop down, after the go button is pressed the relevant information will be displayed on the screen.

Condition(s) tested by second call:

The user will choose the extra information they wish to see about their state from the drop down menu. When the Go! button is pressed the information will appear below the dropdown.

3.d.iii.

Results of the first call:

The result of the first call is the information on the General_Info screen getting populated with the correct information based on the user's selected state.

Results of the second call:

The result of the second call will show the extra information about the state the user requested from the drop down menu.